

Data flow analysis apparatus for debugging, testing and maintenance of source program - has extraction unit which extracts information corresponding to variable other than predetermined procedure from source program

Patent Assignee: NIPPON STEEL CORP (YAWA )

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Main IPC	Week
JP 10289102	A	19981027	JP 9832612	A	19980216	G06F-009/06	199902 B

Priority Applications (No Type Date): JP 9732452 A 19970217

Patent Details:

Patent	Kind	Lan	Pg	Filing Notes	Application	Patent
JP 10289102	A		5			

Abstract (Basic): JP 10289102 A

The apparatus includes a definition information storing file which stores information corresponding to variable definition which regulates predetermined procedure with specific programming language. The source program described in a specific programming language, is read. A procedure judgment unit judges whether predetermined procedure is included.

Based on the judgment result, the information corresponding to variable definition is read out. A definition information extracting unit which extracts the information corresponding to variable from the source program, other than the predetermined procedure. A data flow analysis unit analyses data flow based on information corresponding to variable.

ADVANTAGE - Shortens data flow analysis time.

Dwg.2/3

Title Terms: DATA; FLOW; ANALYSE; APPARATUS; DEBUG; TEST; MAINTAIN; SOURCE; PROGRAM; EXTRACT; UNIT; EXTRACT; INFORMATION; CORRESPOND; VARIABLE; PREDETERMINED; PROCEDURE; SOURCE; PROGRAM

Derwent Class: T01

International Patent Class (Main): G06F-009/06

File Segment: EPI

Manual Codes (EPI/S-X): T01-F06

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-289102

(43) 公開日 平成10年(1998)10月27日

(51) Int.Cl.<sup>5</sup>

G 0 6 F 9/06

識別記号

5 4 0

F I

G 0 6 F 9/06

5 4 0 U

審査請求 未請求 請求項の数 3 O L (全 5 頁)

(21) 出願番号 特願平10-32612

(22) 出願日 平成10年(1998)2月16日

(31) 優先権主張番号 特願平9-32452

(32) 優先日 平9(1997)2月17日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000006655

新日本製鐵株式会社

東京都千代田区大手町2丁目6番3号

(72) 発明者 松井 優美子

東京都千代田区大手町2丁目6番3号 新

日本製鐵株式会社内

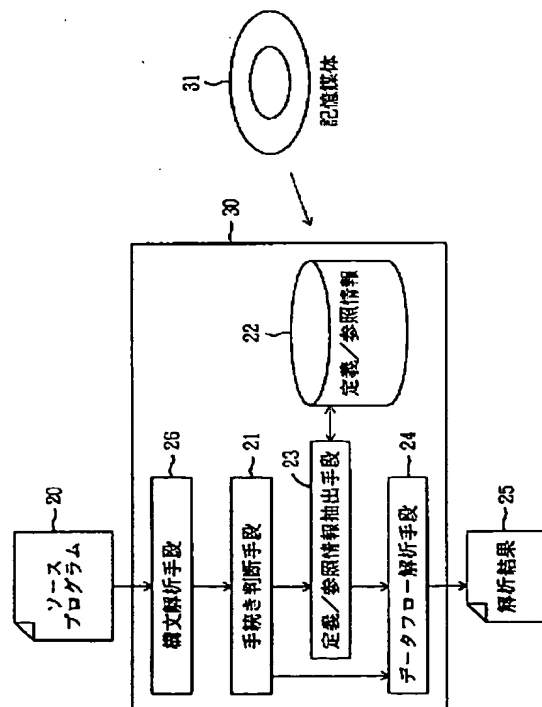
(74) 代理人 弁理士 半田 昌男

(54) 【発明の名称】 プログラムのデータフロー解析装置

(57) 【要約】

【課題】 ソースプログラムの内容を見ることができない手続きを含むプログラムについてもデータフロー解析を可能とし、また、ソースプログラムを見ることができる手続きを含むプログラムであってもそのデータフロー解析に要する時間を短縮することができるプログラムのデータフロー解析装置を提供する。

【解決手段】 C言語の標準ライブラリに含まれる手続きは、その仕様は予め明らかである。このように仕様が明らかになっているライブラリに含まれる各手続きについて、その仕様から、データ依存関係を予め調べておいて、その結果だけを記述した定義／参照情報格納ファイルを用意する。ある手続きの呼び出しがあったらこのテーブルを参照し、「定義」であるのか、「参照」であるのかという情報を抽出する。



**【特許請求の範囲】**

【請求項 1】 特定のプログラミング言語で記述された所定の手続きにおいて規定された変数の定義／参照に関する情報を格納した定義／参照情報格納手段と、前記特定のプログラミング言語で記述されたソースプログラムを読み込んで、前記所定の手続きが含まれているかを判断する手続き判断手段と、

前記手続き判断手段の判断結果に基づいて、前記所定の手続きについては前記定義／参照情報格納手段から変数の定義／参照に関する情報を読み出し、前記所定の手続き以外の手続きについては前記ソースプログラムの内容から変数の定義／参照に関する情報を抽出する定義／参照情報抽出手段と、

前記変数の定義／参照に関する情報に基づいてデータフロー解析を行うデータフロー解析手段と、を有することを特徴とするプログラムのデータフロー解析装置。

【請求項 2】 前記所定の手続きは、実行可能なプログラムを含むものである請求項 1 記載のデータフロー解析装置。

【請求項 3】 コンピュータを、特定のプログラミング言語で記述された所定の手続きにおいて規定された変数の定義／参照に関する情報を格納した定義／参照情報格納手段と、前記特定のプログラミング言語で記述されたソースプログラムを読み込んで、前記所定の手続きが含まれているかを判断する手続き判断手段と、

前記手続き判断手段の判断結果に基づいて、前記所定の手続きについては前記定義／参照情報格納手段から変数の定義／参照に関する情報を読み出し、前記所定の手続き以外の手続きについては前記ソースプログラムの内容から変数の定義／参照に関する情報を抽出する定義／参照情報抽出手段と、

前記変数の定義／参照に関する情報に基づいてデータフロー解析を行うデータフロー解析手段として、機能させるためのプログラムを記録した機械読み取り可能な記憶媒体。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】 本発明は、一般的なプログラム言語で記述されたソースプログラムについてデバッグ、テスト、保守などを行う際に利用されるプログラムのデータフロー解析装置に関する。

**【0002】**

【従来の技術】 プログラムのデータフロー解析は、ソースプログラムを作成した後のデバッグやテストにおいて行われるだけでなく、既に作成されているプログラムを必要に応じて修正し又は改造するといった保守作業を行う際にも行われる。保守作業を行う場合にデータフロー解析を行うことによって、特に大規模なプログラムの修

正等を行う際に、必要な情報を効率よく抽出することができる。かかるデータフロー解析を行う場合に、そのソースプログラムに含まれる各変数の参照について、その値の定義箇所を求め、また、変数の代入についてその値の参照箇所を求める定義／参照解析を行うことが必要である。

**【0003】**

【発明が解決しようとする課題】 ところで、例えば C 言語を始めとする多くの高級言語では、種々の手続きが標準ライブラリ内に用意され、プログラムの中からこの手続きを指定することによって、その手続きを呼び出すことができる。また、ある手続きの中で、また別の手続きを呼び出すことも可能である。しかし、手続きの中でまた別の手続きを呼び出すと、そこでもデータの受渡しが行われるので、データ依存関係を正確に調べようとすると、手続きとして呼び出された手続きの内部についてもデータ依存関係をすべて調べなければならない。このような作業は、特に規模の大きいプログラムの場合においては、実行に長時間を要する。

【0004】 また、標準ライブラリに含まれる手続きが、コンパイルされたあとの実行可能プログラムとして提供されている場合には、そのソースプログラムの内容が存在せず、このような手続きを利用して作成したプログラムについてはデータ依存関係を調べることができなかった。したがって、従来のデータフロー解析装置では、このようなプログラムについてはデータフロー解析を行うことができなかった。

【0005】 本発明は上記事情に基づいてなされたものであり、ソースプログラムの内容を見ることができない手続きを含むプログラムについてもデータフロー解析を可能とし、また、ソースプログラムを見ることができる手続きを含むプログラムであってもそのデータフロー解析に要する時間を短縮することができるプログラムのデータフロー解析装置を提供することを目的とする。

**【0006】**

【課題を解決するための手段】 上記の課題を解決するために、本発明は、特定のプログラミング言語で記述された所定の手続きにおいて規定された変数の定義／参照に関する情報を格納した定義／参照情報格納手段と、前記特定のプログラミング言語で記述されたソースプログラムを読み込んで、前記所定の手続きがどこに含まれているかを判断する手続き判断手段と、前記手続き判断手段の判断結果に基づいて、前記所定の手続きについては前記定義／参照情報格納手段から変数の定義／参照に関する情報を読み出し、前記所定の手続き以外の手続きについては前記ソースプログラムの内容から変数の定義／参照に関する情報を抽出する定義／参照情報抽出手段と、前記変数の定義／参照に関する情報に基づいてデータフロー解析を行うデータフロー解析手段と、を有することを特徴とする。

【0007】本発明は、上記より、定義／参照情報格納手段に、予め所定の手続きについて変数の定義／参照に関する情報を格納する。この場合、所定の手続きには、例えばコンパイルされた後の実行可能なプログラムも含まれる。そして、読み込まれたソースプログラムについて一行ずつ変数の定義／参照に関する情報の抽出を行っている際に、この所定の手続きに至ったら、定義／参照情報格納手段を参照して変数の定義／参照に関する情報を抽出する。このような作業を読み込んだソースプログラム全体について行う。従来は、ソースプログラムがなくて変数の定義／参照に関する情報を抽出することができなかつた手続きについても、かかる情報の抽出が可能となり、その後のデータフロー解析を行うことが可能となる。尚、本明細書では、「手続き」といった場合に、C言語における関数なども含むものとする。

#### 【0008】

【発明の実施の形態】以下に図面を参照して、本発明の一実施形態について説明する。図1は、C言語で記述されたあるソースプログラムの一例を示している。同図において、4行目に「read」という手続きが記述されている。この手続きは、これとは別の手続きである「scanf」を7行目において呼び出している。ここで、「scanf」はC言語の標準ライブラリに含まれる手続きであり、「%d」は、定義する変数*i*を10進数の整数型とする旨の指定である。

【0009】従来のデータフロー解析装置は、単一手続き内のデータフロー解析を行っているので、図1のプログラムのように、一つの手続きの中に別の手続きを呼び出す呼び出し文が含まれているソースプログラムについて、従来の装置でデータフロー解析を行わせようとしても、できない場合があった。このように単一手続きの内部に手続きの呼び出しがある場合には、「手続き間解析」を行うことが一般的であり、これを行うと、単一手続き内の解析に比べて実行速度が大幅に遅くなる。ここで、データフロー解析とは、どの定義がどこで参照されるかを解析することという。

【0010】通常の単一手続き内のデータフロー解析では、4行目の手続き「read」にはデータ依存関係が存在しないことになるが、実際には8行目の「*i*」が、7行目に記述された手続き「scanf」を介して*i*にデータ依存している。これは、C言語の標準ライブラリに含まれる「scanf」の仕様によって、第2引数*i*が定義されるからである。尚、「データ依存」とは、次のことをいう。文S1、文S2、変数*v*があった場合に、

- ①文S1において変数*v*が定義されること
- ②文S2において変数*v*が参照されること
- ③文S1から文S2への実行可能なパスであって、変数*v*を再定義しないパスが存在すること

という3つの条件を満足している場合に、文S2は、変

数*v*に関して文S1にデータ依存するという。

【0011】本実施形態にかかるデータフロー解析装置では、通常の単一手続き内のデータフロー解析と、特定の手続きに関する手続き情報を利用することにより、単一手続き内のデータフロー解析よりも適用可能な条件を広げ、しかも手続き間解析よりも実行速度が早く、かつ正確な解析を行うことを可能とする。上述のように、図1に示すソースプログラムにおいて7行目の「scanf」は、標準ライブラリに含まれる手続きであるため、その仕様は予め明らかになっている。すなわち、図1の例では、*i*という変数が定義されていることが予め分かっている。そこで、この「scanf」のように、仕様が明らかになっている標準ライブラリに含まれる各手続きについて、その仕様から、データ依存関係を予め調べておいて、その結果だけを記述した定義／参照情報格納ファイルを用意する。例えば「scanf」の場合であれば、引数である*i*は、ここで定義される旨の情報を定義／参照情報格納ファイル内に格納しておく。

【0012】また、これとは異なる手続き、例えば後に続く引数を画面に表示する旨の手続きである「printf」という手続きであれば、この引数がこの手続きによって「参照」される。したがって、「printf」という手続きについては、引数がここで参照される旨の情報を定義／参照情報格納ファイル内に格納しておく。このように、定義／参照情報格納ファイルには、多くの手続きについて、予めそれぞれの手続きと、それらが変数の「定義」なのか「参照」なのかを結び付けたテーブルを用意しておく。このなかで、ある手続きの呼び出しがあったならば、このテーブルにその手続きが含まれているかどうかを判断する。呼び出された手続きがこのテーブルにあり、かつ、それが変数の「定義」であるのか、「参照」であるのかという情報を抽出する。

【0013】定義／参照情報格納ファイルに格納する手続きは、C言語の標準ライブラリに含まれるものには限られない。例えば、ある工場で、よく利用される特殊な手続きを作り、その工場で一般的に使われているようなものとなっている場合でも、上記の標準ライブラリに含まれる手続きの場合と同じような定義／参照情報格納ファイルを作成しておけば、同じように正確で効率的な解析を行うことができる。

【0014】次に、図2及び図3を参照して、本実施形態のプログラムのデータフロー解析装置の動作について説明する。図2は、本実施形態のプログラムのデータフロー解析装置のブロック図、図3は、その動作を示すフローチャートである。尚、図2に示した構文解析手段26、手続き判断手段21、定義／参照情報抽出手段23、データフロー解析手段24は、これらの機能をプログラムとして記憶媒体31に記録したものをコンピュータ30が読み取ることによって、コンピュータ30上で実現することができる。したがって、この記憶媒体31

10

20

30

40

50

を例えばフレキシブルディスクやCD-ROMといった取り外し可能なものとすれば、この記憶媒体31を配付することによって、この記憶媒体31を読み取って実行できる任意のコンピュータ上で本実施形態のプログラムのデータフロー解析装置を実現することができる。このとき、定義／参照情報格納ファイル22の内容も、これらのプログラムと共に記憶媒体31に記憶させておいてもよい。

【0015】図3のフローチャートにおいて、まずS10で、構文解析手段26が、入力されるソースプログラム20を元に構文解析木を生成し、生成された構文解析木上から一文を読み込む。S11で、手続き判断手段21はこの文の中に定義／参照情報格納ファイル22に格納された特定の手続きが含まれているか否かを判断し、含まれていれば、S12において、定義／参照情報抽出手段23が定義／参照情報格納ファイル22を参照し、S13において変数が「定義」であるか、「参照」であるかという情報を引き出す。一方、S11において、この文に定義／参照情報格納ファイル22に格納された手続きが含まれていないと判断したときは、S14において、定義／参照情報抽出手段23は変数が「定義」されているか「参照」されているか、あるいはどちらでもないかを判断する。S15において、手続き判断手段21は更にプログラムの文が続いているかどうかを判断し、更に続いていれば、S10に戻り、次の文の情報を読み込んで上記と同様の動作を実行する。また、S15において、最後の文に至ったと判断されると、S16において、データフロー解析手段24がデータフローの解析を実行する。その結果は解析結果25として出力される。

【0016】上記のように、ソースプログラム内に、従来のデータフロー解析装置ではデータ依存を調べることができなかった手続きが含まれていても、予め定義／参照情報格納ファイル22に必要な情報を記述しておき、必要に応じてこれを参照することによって、従来は調べることができなかった手続きや、実際に調べようとすると実行に長時間を要するような手続きについてもデータ

依存関係を短時間で調べることが可能となる。

【0017】尚、本発明は上記実施形態には限定されず、その要旨の範囲内で種々の変更が可能である。例えば上記では、C言語の標準ライブラリに含まれる「scanf」や「printf」といった典型的な手続きについて説明したが、本発明は、C言語におけるこれ以外の手続きもしくは、ユーザーが作成したライブラリに含まれる手続きについても適用することができ、更に、C言語以外の多くの高級言語についても、同様に適用することができる。

【0018】

【発明の効果】以上説明したように、本発明によれば、ソースプログラムの内容を見ることができない手続きを含むプログラムについてもデータフロー解析を可能とし、また、ソースプログラムを見ることができる手続きを含むプログラムであってもそのデータフロー解析に要する時間を短縮することができるプログラムのデータフロー解析装置を提供することができる。

【図面の簡単な説明】

【図1】C言語で記述されたあるソースプログラムの一例を示す図である。

【図2】本発明の一実施形態のプログラムのデータフロー解析装置のブロック図である。

【図3】本発明の一実施形態のプログラムのデータフロー解析装置の動作を示すフローチャートである。

【符号の説明】

20	ソースプログラム
21	手続き判断手段
22	定義／参照情報格納ファイル
23	定義／参照情報抽出手段
24	データフロー解析手段
25	解析結果
26	構文解析手段
30	コンピュータ
31	記憶媒体

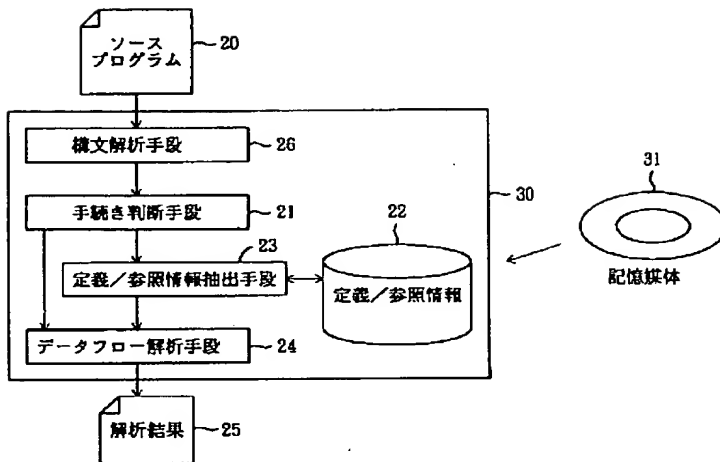
【図1】

```

1  #include <stdio.h>
2
3  int
4  read ()
5  {
6      int i;
7      scanf ("%d", i);
8      return i;
9  }

```

【図 2】



【図 3】

